



Guía para desarrolladores

Amazon DCV Session Manager



Amazon DCV Session Manager: Guía para desarrolladores

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Session Manager?	1
Cómo funciona Session Manager	1
Características	3
Introducción a la API de Session Manager	5
Paso 1: Generar el cliente de la API	5
Paso 2: Registrar la API del cliente	6
Paso 3: Obtener un token de acceso y realizar una solicitud a la API	7
Referencia de la API de Session Manager	10
CloseServers	10
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
CreateSessions	13
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
DescribeServers	21
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
DescribeSessions	32
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
DeleteSessions	39
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
GetSessionConnectionData	41
Parámetros de solicitud	7
Parámetros de respuesta	11
Información adicional	45
Ejemplo	12
GetSessionScreenshots	47

Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
OpenServers	51
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
UpdateSessionPermissions	53
Parámetros de solicitud	7
Parámetros de respuesta	11
Ejemplo	12
Notas de la versión e historial de revisión	56
Notas de la versión	56
2024.0-457: 1 de octubre de 2024	57
2023.1-17652: 1 de agosto de 2024	57
2023.1-16388: 26 de junio de 2024	57
2023.1— 9 de noviembre de 2023	58
2023.0-15065: 4 de mayo de 2023	58
2023.0-14852— 28 de marzo de 2023	58
2022.2-13907— 11 de noviembre de 2022	59
2022.1-13067— 29 de junio de 2022	59
2022.0-11952— 23 de febrero de 2022	59
2021.3-11591— 20 de diciembre de 2021	60
2021.2-11445— 18 de noviembre de 2021	60
2021.2-11190— 11 de octubre de 2021	60
2021.2-11042: 1 de septiembre de 2021	61
2021.1-10557— 31 de mayo de 2021	61
2021.0-10242— 12 de abril de 2021	62
2020.2-9662— 04 de diciembre de 2020	63
.....	63
Historial de documentos	63

¿Qué es Amazon DCV Session Manager?

Note

Amazon DCV se conocía anteriormente como NICE DCV.

Amazon DCV Session Manager es un conjunto de paquetes de software instalables (un agente y un broker) y una interfaz de programación de aplicaciones (API) que facilita a los desarrolladores y proveedores de software independientes (ISV) la creación de aplicaciones front-end que crean y gestionan mediante programación el ciclo de vida de las sesiones de Amazon DCV en una flota de servidores Amazon DCV.

Esta guía explica cómo utilizar las API de Session Manager para gestionar el ciclo de vida de las sesiones de Amazon DCV. Para obtener más información sobre cómo instalar y configurar el broker y el agente de Session Manager, consulte la [Guía del administrador de Amazon DCV Session Manager](#).

Requisitos previos

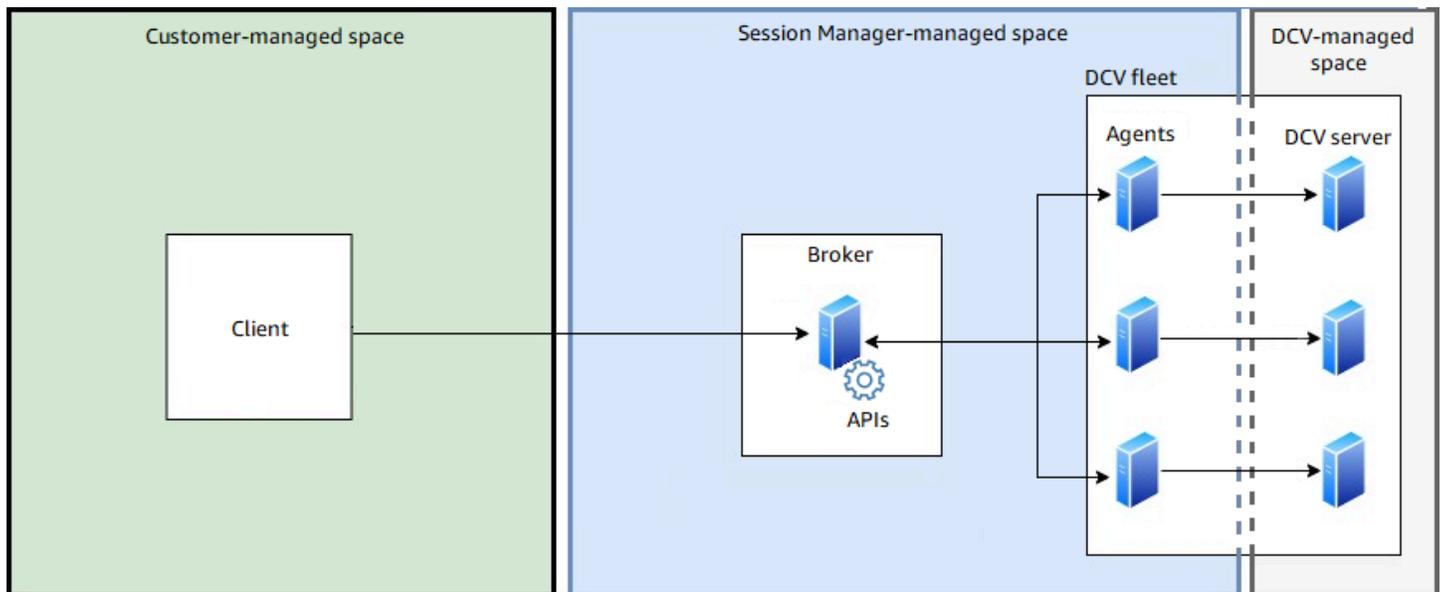
Antes de empezar a trabajar con las API de Session Manager, asegúrese de estar familiarizado con Amazon DCV y las sesiones de Amazon DCV. Para obtener más información, consulte la [Guía del administrador de Amazon DCV](#).

Temas

- [Cómo funciona Session Manager](#)
- [Características](#)

Cómo funciona Session Manager

En el siguiente diagrama se muestran los componentes de alto nivel de Session Manager.



Broker

El broker es un servidor web que aloja y expone las API de Session Manager. Recibe y procesa solicitudes de API para gestionar las sesiones de Amazon DCV del cliente y, a continuación, pasa las instrucciones a los agentes correspondientes. El broker debe estar instalado en un host distinto de los servidores Amazon DCV, pero debe ser accesible para el cliente y debe poder acceder a los agentes.

Agente

El agente está instalado en cada servidor Amazon DCV de la flota. Los agentes reciben instrucciones del broker y las ejecutan en sus servidores Amazon DCV correspondientes. Los agentes también supervisan el estado de los servidores Amazon DCV y envían actualizaciones de estado periódicas al broker.

API

Session Manager expone un conjunto de interfaces de programación de aplicaciones (API) REST que se pueden utilizar para administrar las sesiones de Amazon DCV en una flota de servidores Amazon DCV. El broker aloja y expone las API. Los desarrolladores pueden crear clientes de administración de sesiones personalizados que llamen a las API.

Cliente

El cliente es la aplicación o el portal front-end que se desarrolla para llamar a las API de Session Manager expuestas por el broker. Los usuarios finales utilizan el cliente para gestionar las sesiones alojadas en los servidores Amazon DCV de la flota.

Token de acceso

Para realizar una solicitud de API, debe proporcionar un token de acceso. Las API de cliente registradas pueden solicitar tokens al broker o a un servidor de autorización externo. Para solicitar el token y acceder a él, la API del cliente debe proporcionar credenciales válidas.

API del cliente

La API del cliente se genera a partir del archivo YAML de definición de la API de Session Manager, utilizando Swagger Codegen. La API del cliente se utiliza para realizar solicitudes de API.

Sesión de Amazon DCV

Debe crear una sesión de Amazon DCV en el servidor Amazon DCV al que puedan conectarse sus clientes. Los clientes solo pueden conectarse a un servidor Amazon DCV si hay una sesión activa. Amazon DCV admite sesiones virtuales y de consola. Las API de Session Manager sirven para gestionar el ciclo de vida de las sesiones de Amazon DCV. Las sesiones de Amazon DCV pueden tener uno de los siguientes estados:

- CREATING: el broker está creando la sesión.
- READY: la sesión está lista para aceptar conexiones de clientes.
- DELETING: se está eliminando la sesión.
- DELETED: se ha eliminado la sesión.
- UNKNOWN: no se puede determinar el estado de la sesión. Es posible que el agente y el broker no puedan comunicarse.

Características

DCV Session Manager ofrece las siguientes características:

- Proporciona información sobre sesiones de Amazon DCV: obtenga información sobre las sesiones que se ejecutan en varios servidores Amazon DCV.
- Gestione el ciclo de vida de varias sesiones de Amazon DCV: cree o elimine distintas sesiones para varios usuarios en múltiples servidores Amazon DCV con una sola solicitud de API.
- Admite etiquetas: utilice etiquetas personalizadas para identificar un grupo de servidores Amazon DCV al crear sesiones.

- Administra los permisos para varias sesiones de Amazon DCV: modifique los permisos de usuario para varias sesiones con una solicitud de API.
- Proporciona información de conexión: recupera la información de conexión del cliente para las sesiones de Amazon DCV.
- Compatible con servidores en las instalaciones y en la nube: utilice Session Manager en servidores de AWS, en las instalaciones o alternativos basados en la nube.

Introducción a la API de Session Manager

La API de Amazon DCV Session Manager proporciona una interfaz automatizada para gestionar sesiones de escritorio remoto. A través de esta API, los desarrolladores pueden crear, enumerar, iniciar, detener y controlar las sesiones de DCV mediante programación. Esto permite integrar la funcionalidad de Amazon DCV en aplicaciones y flujos de trabajo personalizados. Al aprovechar esta API, las organizaciones pueden optimizar la administración de las cargas de trabajo de visualización remota y automatizar muchas tareas comunes.

Antes de empezar a realizar llamadas a la API de Amazon DCV, necesitará obtener un token de acceso que autentique su aplicación y la autorice a acceder a los recursos necesarios. La API de Amazon DCV utiliza OAuth 2.0 para la autenticación, por lo que tendrá que registrar la aplicación y recuperar las credenciales necesarias. Cuando tenga su token de acceso, podrá empezar a enviar solicitudes a los puntos de conexión de la API de Amazon DCV para empezar a procesar los datos.

Temas

- [Paso 1: Generar el cliente de la API](#)
- [Paso 2: Registrar la API del cliente](#)
- [Paso 3: Obtener un token de acceso y realizar una solicitud a la API](#)

Paso 1: Generar el cliente de la API

Las API de Session Manager se definen en un único archivo YAML. Las API se basan en la especificación OpenAPI3.0, que define una interfaz estándar e independiente del idioma para las API RESTful. Para obtener más información, consulte [Especificación de OpenAPI](#).

Con el archivo YAML, puede generar un cliente de API en uno de los lenguajes compatibles. Para ello, debe utilizar Swagger Codegen 3.0 o posterior. Para obtener más información acerca de los idiomas compatibles, consulte el [repositorio swagger-codegen](#).

Para generar el cliente de la API

1. Descargue el archivo YAML de la API de Session Manager desde el broker de Session Manager. El archivo YAML está disponible en la siguiente URL.

```
https://broker_host_ip:port/dcv-session-manager-api.yaml
```

2. Instale Swagger Codegen.

- macOS

```
$ brew install swagger-codegen
```

- Otras plataformas

```
$ git clone https://github.com/swagger-api/swagger-codegen --branch 3.0.0
```

```
$ cd swagger-codegen
```

3. Genere el cliente de la API.

- macOS

```
$ swagger-codegen generate -i /path_to/yaml_file -l language -o $output_folder
```

- Otras plataformas

```
$ mvn clean package
```

```
$ java -jar modules/swagger-codegen-cli/target/swagger-codegen-cli.jar generate -i /path_to/yaml_file -l language -o output_folder
```

Paso 2: Registrar la API del cliente

Las solicitudes de API utilizan un token de acceso para verificar sus credenciales. Estas credenciales se basan en un ID de cliente y una contraseña de cliente que se genera cuando su cliente se registra en el broker.

Para acceder a este token, debe registrarse en el broker. Utilice [register-api-client](#) para registrar la API del cliente.

Si no tiene un ID de cliente y una contraseña de cliente, debe solicitarlos al administrador del broker.

Paso 3: Obtener un token de acceso y realizar una solicitud a la API

En este ejemplo, se explican los pasos necesarios para configurar el token de acceso y, a continuación, se muestra cómo realizar una solicitud básica a la API. Esto le proporcionará los conocimientos básicos para empezar a crear aplicaciones más avanzadas impulsadas por la API de Amazon DCV.

En esta sección le mostraremos cómo hacerlo utilizando la API `DescribeSessions` como ejemplo.

Example

Primero importamos los modelos necesarios para la aplicación.

A continuación, declaramos las variables para el ID del cliente (`__CLIENT_ID`), la contraseña del cliente (`__CLIENT_SECRET`) y la URL del broker, incluido el número de puerto (`__PROTOCOL_HOST_PORT`).

A continuación, creamos una función llamada `build_client_credentials` que genera las credenciales del cliente. Para generar las credenciales del cliente, primero debe concatenar el ID y la contraseña del cliente, separar los valores con dos puntos (`client_id:client_password`) y, a continuación, codificar en Base64 toda la cadena.

```
import swagger_client
import base64
import requests
import json
from swagger_client.models.describe_sessions_request_data import
    DescribeSessionsRequestData
from swagger_client.models.key_value_pair import KeyValuePair
from swagger_client.models.delete_session_request_data import DeleteSessionRequestData
from swagger_client.models.update_session_permissions_request_data import
    UpdateSessionPermissionsRequestData
from swagger_client.models.create_session_request_data import CreateSessionRequestData

__CLIENT_ID = '794b2dbb-bd82-4707-a2f7-f3d9899cb386'
__CLIENT_SECRET = 'MzcxNzJhN2UtYjEzNS00MjN2YtMjF1ZmRlZWJmDU1'
__PROTOCOL_HOST_PORT = 'https://<broker-hostname>:8443'

def build_client_credentials():
```

```
client_credentials = '{client_id}:{client_secret}'.format(client_id=__CLIENT_ID,
client_secret=__CLIENT_SECRET)
return base64.b64encode(client_credentials.encode('utf-8')).decode('utf-8')
```

Ahora que tenemos nuestras credenciales del cliente, podemos emplearlas para solicitar un token de acceso al broker. Para ello, creamos una función llamada `get_access_token`. Debe llamar a un POST en `https://Broker_IP:8443/oauth2/token?grant_type=client_credentials` y proporcionar un encabezado de autorización que incluya las credenciales del cliente codificadas en Basic y un tipo de contenido de `application/x-www-form-urlencoded`.

```
def get_access_token():
    client_credentials = build_client_credentials()
    headers = {
        'Authorization': 'Basic {}'.format(client_credentials),
        'Content-Type': 'application/x-www-form-urlencoded'
    }
    endpoint = __PROTOCOL_HOST_PORT + '/oauth2/token?grant_type=client_credentials'
    print('Calling', endpoint, 'using headers', headers)
    res = requests.post(endpoint, headers=headers, verify=True)
    if res.status_code != 200:
        print('Cannot get access token:', res.text)
        return None
    access_token = json.loads(res.text)['access_token']
    print('Access token is', access_token)
    return access_token
```

Ahora creamos las funciones necesarias para crear una instancia de una API de cliente. Para crear una instancia de una API de cliente debe especificar la configuración del cliente y los encabezados que se utilizarán en las solicitudes. La función `get_client_configuration` crea un objeto de configuración que incluye la dirección IP y el puerto del Agente y la ruta del certificado autofirmado del broker, que debería haber recibido del administrador del broker. La función `set_request_headers` crea un objeto de encabezado de solicitud que incluye las credenciales del cliente y el token de acceso.

```
def get_client_configuration():
    configuration = swagger_client.Configuration()
    configuration.host = __PROTOCOL_HOST_PORT
    configuration.verify_ssl = True
    # configuration.ssl_ca_cert = cert_file.pem
```

```
return configuration

def set_request_headers(api_client):
    access_token = get_access_token()
    api_client.set_default_header(header_name='Authorization',
                                  header_value='Bearer {}'.format(access_token))

def get_sessions_api():
    api_instance =
    swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance
```

Por último, creamos un método principal que llama a la API `DescribeSessions`. Para obtener más información, consulte [DescribeSessions](#).

```
def describe_sessions(session_ids=None, next_token=None, tags=None, owner=None):
    filters = list()
    if tags:
        for tag in tags:
            filter_key_value_pair = KeyValuePair(key='tag:' + tag['Key'],
value=tag['Value'])
            filters.append(filter_key_value_pair)
    if owner:
        filter_key_value_pair = KeyValuePair(key='owner', value=owner)
        filters.append(filter_key_value_pair)

    request = DescribeSessionsRequestData(session_ids=session_ids, filters=filters,
next_token=next_token)
    print('Describe Sessions Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.describe_sessions(body=request)
    print('Describe Sessions Response', api_response)

def main():
    describe_sessions(
        session_ids=['SessionId1895', 'SessionId1897'],
        owner='an owner 1890',
        tags=[{'Key': 'ram', 'Value': '4gb'}])
```

Referencia de la API de Session Manager

Esta referencia proporciona detalles sobre las acciones de la API disponibles, los parámetros necesarios y los formatos de respuesta para que pueda aprovechar eficazmente la API de Session Manager en sus propios sistemas. Con la API de Session Manager, puede iniciar, detener y obtener detalles sobre las sesiones interactivas. Esto le permite automatizar e integrar la funcionalidad en sus aplicaciones y flujos de trabajo.

Temas

- [CloseServers](#)
- [CreateSessions](#)
- [DescribeServers](#)
- [DescribeSessions](#)
- [DeleteSessions](#)
- [GetSessionConnectionData](#)
- [GetSessionScreenshots](#)
- [OpenServers](#)
- [UpdateSessionPermissions](#)

CloseServers

Cierra uno o más servidores Amazon DCV. Cuando cierra un servidor Amazon DCV, deja de estar disponible para la ubicación de sesiones de Amazon DCV. No puede crear sesiones de Amazon DCV en servidores cerrados. Al cerrar un servidor se garantiza que no hay sesiones ejecutándose en él y que los usuarios no pueden crear nuevas sesiones en él.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

ServerId

ID del túnel que se va a cerrar.

Tipo: cadena

Obligatorio: sí

Force

Fuerza la operación de cierre. Si especifica `true`, el servidor se cerrará aunque tenga sesiones en ejecución. Las sesiones siguen ejecutándose.

Tipo: Booleano

Requerido: no

Parámetros de respuesta

RequestId

ID único de la solicitud.

SuccessfulList

Información sobre los servidores Amazon DCV que se cerraron correctamente. Esta estructura de datos incluye el siguiente parámetro de respuesta anidado:

ServerId

ID del servidor que se cerró correctamente.

UnsuccessfulList

Información sobre los servidores Amazon DCV que no se han podido cerrar. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

CloseServerRequestData

Información sobre la solicitud original que falló. Esta estructura de datos incluye el siguiente parámetro de respuesta anidado:

ServerId

ID del servidor Amazon DCV que no se ha podido cerrar.

Force

Parámetro forzar solicitado.

FailureCode

Código del error.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

El siguiente ejemplo cierra dos servidores Amazon DCV (`serverId1` y `serverId2`). El servidor `serverId2` no existe y se produce un error.

```
from swagger_client.models import CloseServerRequestData

def get_servers_api():
    api_instance =
    swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def close_servers(server_ids):
    request = [CloseServerRequestData(server_id=server_id) for server_id in
server_ids]
    print('Close Servers Request:', request)
    api_instance = get_servers_api()
    api_response = api_instance.close_servers(body=request)
    print('Close Servers Response:', api_response)
    open_servers(server_ids)

def main():
    close_servers(["serverId1", "serverId2"])
```

Respuesta

A continuación se muestra la salida de ejemplo.

```
{
  "RequestId": "4d7839b2-a03c-4b34-a40d-06c8b21099e6",
  "SuccessfulList": [
    {
      "ServerId": "serverId1"
    }
  ],
  "UnsuccessfulList": [
    {
      "OpenServerRequestData": {
        "ServerId": "serverId2"
      },
      "FailureCode": "DCV_SERVER_NOT_FOUND",
      "FailureReason": "Dcv server not found."
    }
  ]
}
```

CreateSessions

Crea una nueva sesión de Amazon DCV con los detalles especificados.

Acciones de API

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

Name

Nombre de la sesión.

Tipo: cadena

Obligatorio: sí

Owner

Nombre del propietario de la sesión. Debe ser el nombre de un usuario existente en el servidor Amazon DCV de destino.

Tipo: cadena

Obligatorio: sí

Type

El tipo de sesión. Para obtener más información sobre los tipos de sesiones, consulte [Introducción a las sesiones Amazon DCV](#) en la Guía del administrador de Amazon DCV.

Valores válidos: CONSOLE | VIRTUAL

Tipo: cadena

Obligatorio: sí

InitFile

Se admite con sesiones virtuales en servidores Amazon DCV de Linux. No se admite con sesiones de consola en servidores Amazon DCV de Windows y Linux. Ruta del script personalizado del servidor Amazon DCV que se va a ejecutar para inicializar la sesión cuando se cree. La ruta del archivo es relativa al directorio de inicio especificado para el parámetro de configuración del agente `agent.init_folder`. Si el archivo se encuentra en el directorio de inicio especificado, especifique únicamente el nombre del archivo. Si el archivo no se encuentra en el directorio de inicio especificado, especifique la ruta relativa. Para obtener más información, consulte [Archivo de configuración del agente](#) en la Guía del administrador de Amazon DCV Session Manager.

Tipo: cadena

Requerido: no

MaxConcurrents

Número máximo de clientes Amazon DCV simultáneos.

Tipo: entero

Requerido: no

DcvGLEnabled

Indica si la sesión virtual está configurada para utilizar OpenGL basado en hardware. Solo se admite con sesiones virtuales. Este parámetro no es compatible con los servidores Amazon DCV de Windows.

Valores válidos: true | false

Tipo: Booleano

Requerido: no

PermissionsFile

Contenido codificado en base64 del archivo de permisos. El valor predeterminado son los valores predeterminados del servidor si se omite. Para obtener más información, consulte [Configuración de la autorización de Amazon DCV](#) en la Guía del administrador de Amazon DCV.

Tipo: cadena

Requerido: no

EnqueueRequest

Indica si se debe poner en cola la solicitud si no se puede tramitar inmediatamente.

Tipo: Booleano

Predeterminado: false

Requerido: no

AutorunFile

Compatible con sesiones de consola en servidores Amazon DCV de Windows y sesiones virtuales en servidores Amazon DCV de Linux. No se admite con sesiones de consola en servidores Amazon DCV de Linux.

Ruta de un archivo en el servidor host que se va a ejecutar dentro de la sesión. La ruta del archivo es relativa al directorio especificado para el parámetro de configuración del agente de `agent.autorun_folder`. Si el archivo se encuentra en el directorio de ejecución automática especificado, especifique únicamente el nombre del archivo. Si el archivo no se encuentra en el directorio de inicio especificado, especifique la ruta relativa. Para obtener más información, consulte [Archivo de configuración del agente](#) en la Guía del administrador de Amazon DCV Session Manager.

El archivo se ejecuta en nombre del Propietario especificado. El propietario especificado debe tener permiso para ejecutar el archivo en el servidor. En servidores Amazon DCV de Windows, el archivo se ejecuta cuando el propietario inicia sesión. En servidores Amazon DCV de Linux, el archivo se ejecuta cuando se crea la sesión.

Tipo: cadena

Requerido: no

AutorunFileArguments

Se admite con sesiones virtuales en servidores Amazon DCV de Linux. No se admite en sesiones de consola de servidores Amazon DCV de Windows y Linux. Los argumentos de la línea de comandos se pasan a AutorunFile tras su ejecución en la sesión. Los argumentos se pasan en el orden en que aparecen en la matriz dada. Se puede configurar el número máximo permitido de argumentos y la longitud máxima permitida de cada argumento. Para obtener más información, consulte [Archivo de configuración del broker](#) en la Guía del administrador de Amazon DCV Session Manager.

Tipo: matriz de cadenas

Requerido: no

DisableRetryOnFailure

Indica si no se debe volver a intentar la solicitud de creación de sesión cuando se produce un error en un host Amazon DCV por cualquier motivo. Para obtener más información sobre el mecanismo de reintento de creación de sesión, consulte el [archivo de configuración del broker](#) en la Guía del administrador de Amazon DCV Session Manager.

Tipo: Booleano

Predeterminado: false

Requerido: no

Requirements

Requisitos que debe cumplir el servidor para colocar la sesión. Los requisitos pueden incluir etiquetas del servidor y/o propiedades del servidor; tanto las etiquetas como las propiedades del servidor se recuperan llamando a la API DescribeServers.

Expresiones de condiciones de requisitos:

- $a \neq b$ true si a es distinto de b

- $a = b$ true si a es distinto de b
- $a > b$ true si a es mayor que b
- $a \geq b$ true si a es igual o mayor que b
- $a < b$ true si a es menor que b
- $a \leq b$ true si a es igual o menor que b
- $a = b$ true si a contiene la cadena b

Operadores booleanos de requisitos:

- a y b true si a y b son true
- a o b true si a o b son true
- not a true si a es false

Las claves de las etiquetas deben ir precedidas por `tag:` y las propiedades del servidor deben ir precedidas por `server:`. Las expresiones de requisitos admiten paréntesis `()`.

Ejemplos de requisitos:

- `tag:color = 'pink' and (server:Host.Os.Family = 'windows' or tag:color := 'red')`
- `"server:Host.Aws.Ec2InstanceType := 't2' and server:Host.CpuInfo.NumberOfCpus >= 2"`

Los valores numéricos se pueden especificar mediante la notación exponencial, por ejemplo: `"server:Host.Memory.TotalBytes > 1024E6"`.

Las propiedades admitidas del servidor son:

- `Id`
- `Hostname`
- `Version`
- `SessionManagerAgentVersion`
- `Host.Os.BuildNumber`
- `Host.Os.Family`
- `Host.Os.KernelVersion`
- `Host.Os.Name`
- `Host.Os.Version`

- `Host.Memory.TotalBytes`
- `Host.Memory.UsedBytes`
- `Host.Swap.TotalBytes`
- `Host.Swap.UsedBytes`
- `Host.CpuLoadAverage.OneMinute`
- `Host.CpuLoadAverage.FiveMinutes`
- `Host.CpuLoadAverage.FifteenMinutes`
- `Host.Aws.Ec2InstanceId`
- `Host.Aws.Ec2InstanceType`
- `Host.Aws.Region`
- `Host.Aws.Ec2ImageId`
- `Host.CpuInfo.Architecture`
- `Host.CpuInfo.ModelName`
- `Host.CpuInfo.NumberOfCpus`
- `Host.CpuInfo.PhysicalCoresPerCpu`
- `Host.CpuInfo.Vendor`

Tipo: cadena

Requerido: no

StorageRoot

Especifica la ruta a la carpeta usada para el almacenamiento de sesión. Para obtener más información sobre el almacenamiento de sesiones de Amazon DCV, consulte [Habilitación de almacenamiento de sesión](#) en la Guía del administrador de Amazon DCV.

Tipo: cadena

Requerido: no

Parámetros de respuesta

Id

ID exclusivo de la sesión.

Name

El nombre de la sesión.

Owner

Propietario de la sesión.

Type

Tipo de sesión.

State

Estado de la sesión. Si la solicitud se completa correctamente, la sesión pasa al estado CREATING.

Substate

Subestado de la sesión. Si la solicitud se completa correctamente, el subestado pasa al subestado SESSION_PLACING.

Ejemplo

Python

Solicitud

El siguiente ejemplo crea tres sesiones.

```
from swagger_client.models.create_session_request_data import
    CreateSessionRequestData

def get_sessions_api():
    api_instance =
    swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def create_sessions(sessions_to_create):
    create_sessions_request = list()
    for name, owner, session_type, init_file_path, autorun_file,
    autorun_file_arguments, max_concurrent_clients,\
```

```

        dcv_gl_enabled, permissions_file, requirements, storage_root in
sessions_to_create:
    a_request = CreateSessionRequestData(
        name=name, owner=owner, type=session_type,
        init_file_path=init_file_path, autorun_file=autorun_file,
autorun_file_arguments=autorun_file_arguments,
max_concurrent_clients=max_concurrent_clients,
        dcv_gl_enabled=dcv_gl_enabled, permissions_file=permissions_file,
requirements=requirements, storage_root=storage_root)
    create_sessions_request.append(a_request)

    api_instance = get_sessions_api()
    print('Create Sessions Request:', create_sessions_request)
    api_response = api_instance.create_sessions(body=create_sessions_request)
    print('Create Sessions Response:', api_response)

def main():
    create_sessions([
        ('session1', 'user1', 'CONSOLE', None, None, None, 1, None, '/dcv/
permissions.file', "tag:os = 'windows' and server:Host.Memory.TotalBytes > 1024", "/
storage/root"),
        ('session2', 'user1', 'VIRTUAL', None, 'myapp.sh', None, 1, False, None, "tag:os
= 'linux'", None),
        ('session3', 'user1', 'VIRTUAL', '/dcv/script.sh', 'myapp.sh', ['argument1',
'argument2'], 1, False, None, "tag:os = 'linux'", None),
    ])

```

Respuesta

A continuación se muestra la salida de ejemplo.

```

{
    "RequestId": "e32d0b83-25f7-41e7-8c8b-e89326ecc87f",
    "SuccessfulList": [
        {
            "Id": "78b45deb-1163-46b1-879b-7d8fcbe9d9d6",
            "Name": "session1",
            "Owner": "user1",
            "Type": "CONSOLE",
            "State": "CREATING"
        },
        {
            "Id": " a0c743c4-9ff7-43ce-b13f-0c4d55a268dd",
            "Name": "session2",

```

```
    "Owner": "user1",
    "Type": "VIRTUAL",
    "State": "CREATING"
  },
  {
    "Id": " 10311636-df90-4cd1-bcf7-474e9675b7cd",
    "Name": "session3",
    "Owner": "user1",
    "Type": "VIRTUAL",
    "State": "CREATING"
  }
],
"UnsuccessfulList": [
]
}
```

DescribeServers

Describe uno o varios servidores Amazon DCV.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

ServerIds

ID de los servidores Amazon DCV que se van a describir. Si no se especifica un ID, todos los servidores se devuelven en salida paginada.

Tipo: matriz de cadenas

Requerido: no

NextToken

Token que se utilizará para recuperar la siguiente página de resultados.

Tipo: cadena

Requerido: no

MaxResults

Número máximo de resultados que va a devolver la solicitud en salida paginada. Cuando se utiliza este parámetro, la solicitud devuelve solo el número especificado de resultados en una sola página junto con un elemento de respuesta de NextToken. Los resultados restantes de la solicitud inicial se pueden ver enviando otra solicitud de con el valor NextToken devuelto.

Rango válido: 1-1000

Predeterminado: 1000

Tipo: entero

Requerido: no

Parámetros de respuesta

RequestId

ID único de la solicitud.

Servers

Información acerca de los servidores Amazon DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Id

ID exclusivo del servidor Amazon DCV.

Ip

Dirección IP del servidor Amazon DCV.

Hostname

Nombre de host del servidor Amazon DCV.

Endpoints

Información sobre los puntos de conexión del servidor Amazon DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

Puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Los valores posibles son:

- HTTP: el punto de conexión utiliza el protocolo WebSocket (TCP).
- QUIC: el punto de conexión utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del punto de conexión del servidor. Disponible solo para el protocolo HTTP.

Version

Versión del servidor Amazon DCV.

SessionManagerAgentVersion

Versión del agente de Session Manager que se ejecuta en el servidor Amazon DCV.

Availability

Disponibilidad del servidor Amazon DCV. Los valores posibles son:

- AVAILABLE: el servidor está disponible y listo para la colocación de sesiones.
- UNAVAILABLE: el servidor no está disponible y no puede aceptar la colocación de sesiones.

UnavailabilityReason

Motivo de la falta de disponibilidad del servidor Amazon DCV. Los valores posibles son:

- SERVER_FULL: el servidor Amazon DCV ha alcanzado el número máximo de sesiones simultáneas que puede ejecutar.
- SERVER_CLOSED: el servidor Amazon DCV ha dejado de estar disponible utilizando la API CloseServer.
- UNREACHABLE_AGENT: el broker de Session Manager no puede comunicarse con el agente de Session Manager en el servidor Amazon DCV.
- UNHEALTHY_DCV_SERVER: el agente de Session Manager no se puede comunicar con el servidor Amazon DCV.

- `EXISTING_LOGGED_IN_USER`: (solo servidores Amazon DCV de Windows) un usuario ha iniciado la sesión actual en el servidor Amazon DCV mediante RDP.
- `UNKNOWN`: el broker de Session Manager no puede determinar el motivo.

ConsoleSessionCount

Número de sesiones de consola en el servidor Amazon DCV.

VirtualSessionCount

Número de sesiones virtuales en el servidor Amazon DCV.

Host

Información sobre el servidor host en el que se ejecuta el servidor Amazon DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Os

Información sobre el sistema operativo del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Family

La familia de sistemas operativos. Los valores posibles son:

- `windows`: el servidor host ejecuta un sistema operativo Windows.
- `linux`: el servidor host ejecuta un sistema operativo Linux.

Name

Nombre del sistema operativo.

Version

Versión del sistema operativo.

KernelVersion

(Solo Linux) Versión del kernel del sistema operativo.

BuildNumber

(Solo Windows) Número de compilación del sistema operativo.

Memory

Información sobre la memoria del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

TotalBytes

Memoria total, en bytes, del servidor host.

UsedBytes

Memoria usada, en bytes, del servidor host.

Swap

Información sobre la memoria de intercambio del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

TotalBytes

Tamaño de archivo de intercambio total, en bytes, del servidor host.

UsedBytes

Tamaño de archivo de intercambio usado, en bytes, del servidor host.

Aws

Solo para servidores Amazon DCV que se ejecutan en una instancia de Amazon EC2. Información específica de AWS. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Region

Región de AWS de la instancia Amazon EC2.

Ec2InstanceType

Tipo de instancia de Amazon EC2.

Ec2InstanceId

ID de la instancia de Amazon EC2.

Ec2ImageId

ID de la imagen de Amazon EC2.

CpuInfo

Información sobre la CPU del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Vendor

Proveedor de la CPU del servidor host.

ModelName

Modelo de la CPU del servidor host.

Architecture

Arquitectura de la CPU del servidor host.

NumberOfCpus

Número de CPU del servidor host.

PhysicalCorePerCpu

Número de núcleos de CPU por CPU.

CpuLoadAverage

Información sobre la carga de la CPU del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

OneMinute

Carga media de la CPU durante el último período de 1 minuto.

FiveMinutes

Carga media de la CPU durante el último período de 5 minutos.

FifteenMinutes

Carga media de la CPU durante el último período de 15 minutos.

Gpus

Información sobre las GPU del servidor host. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Vendor

Proveedor de la GPU del servidor host.

ModelName

Modelo de la GPU del servidor host.

LoggedInUsers

Usuarios que han iniciado sesión actualmente en el servidor host. Esta estructura de datos incluye el siguiente parámetro de respuesta anidado:

Username

Nombre de usuario del usuario que ha iniciado sesión.

Tags

Etiquetas asignadas al servicio. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Ejemplo

Python

Solicitud

En el siguiente ejemplo se describen todos los servidores Amazon DCV disponibles. Los resultados se pagan para mostrar dos resultados por página.

```
from swagger_client.models.describe_servers_request_data import
    DescribeServersRequestData

def get_servers_api():
    api_instance =
    swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def describe_servers(server_ids=None, next_token=None, max_results=None):
    request = DescribeServersRequestData(server_ids=server_ids,
    next_token=next_token, max_results=max_results)
    print('Describe Servers Request:', request)
    api_instance = get_servers_api()
    api_response = api_instance.describe_servers(body=request)
    print('Describe Servers Response', api_response)

def main():
```

```
describe_servers(max_results=2)
```

Respuesta

A continuación se muestra la salida de ejemplo.

```
{
  "RequestId": "request-id-123",
  "Servers": [
    {
      "Id": "ServerId123",
      "Ip": "1.1.1.123",
      "Hostname": "node001",
      "DefaultDnsName": "node001",
      "Endpoints": [
        {
          "IpAddress": "x.x.x.x",
          "Port": 8443,
          "WebUrlPath": "/",
          "Protocol": "HTTP"
        }
      ],
      "Version": "2021.0.10000",
      "SessionManagerAgentVersion": "2021.0.300",
      "Availability": "UNAVAILABLE",
      "UnavailabilityReason": "SERVER_FULL",
      "ConsoleSessionCount": 1,
      "VirtualSessionCount": 0,
      "Host": {
        "Os": {
          "Family": "windows",
          "Name": "Windows Server 2016 Datacenter",
          "Version": "10.0.14393",
          "BuildNumber": "14393"
        },
        "Memory": {
          "TotalBytes": 8795672576,
          "UsedBytes": 1743886336
        },
        "Swap": {
          "TotalBytes": 0,
          "UsedBytes": 0
        },
        "Aws": {
```

```
        "Region": "us-west-2b",
        "EC2InstanceType": "t2.large",
        "EC2InstanceId": "i-123456789",
        "EC2ImageId": "ami-12345678987654321"
    },
    "CpuInfo": {
        "Vendor": "GenuineIntel",
        "ModelName": "Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz",
        "Architecture": "x86_64",
        "NumberOfCpus": 2,
        "PhysicalCoresPerCpu": 3
    },
    "CpuLoadAverage": {
        "OneMinute": 0.04853546,
        "FiveMinutes": 0.21060601,
        "FifteenMinutes": 0.18792416
    },
    "Gpus": [],
    "LoggedInUsers": [
        {
            "Username": "Administrator"
        }
    ],
    "Tags": [
        {
            "Key": "color",
            "Value": "pink"
        },
        {
            "Key": "dcv:os-family",
            "Value": "windows"
        },
        {
            "Key": "size",
            "Value": "small"
        },
        {
            "Key": "dcv:max-virtual-sessions",
            "Value": "0"
        }
    ]
},
{
```

```
"Id": "server-id-12456897",
"Ip": "1.1.1.145",
"Hostname": "node002",
"DefaultDnsName": "node002",
"Endpoints": [
  {
    "IpAddress": "x.x.x.x",
    "Port": 8443,
    "WebUrlPath": "/",
    "Protocol": "HTTP"
  },
  {
    "IpAddress": "x.x.x.x",
    "Port": 8443,
    "Protocol": "QUIC"
  }
],
"Version": "2021.0.10000",
"SessionManagerAgentVersion": "2021.0.0",
"Availability": "AVAILABLE",
"ConsoleSessionCount": 0,
"VirtualSessionCount": 5,
"Host": {
  "Os": {
    "Family": "linux",
    "Name": "Amazon Linux",
    "Version": "2",
    "KernelVersion": "4.14.203-156.332.amzn2.x86_64"
  },
  "Memory": {
    "TotalBytes": 32144048128,
    "UsedBytes": 2184925184
  },
  "Swap": {
    "TotalBytes": 0,
    "UsedBytes": 0
  },
  "Aws": {
    "Region": "us-west-2a",
    "EC2InstanceType": "g3s.xlarge",
    "EC2InstanceId": "i-123456789",
    "EC2ImageId": "ami-12345678987654321"
  },
  "CpuInfo": {
```

```
        "Vendor": "GenuineIntel",
        "ModelName": "Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz",
        "Architecture": "x86_64",
        "NumberOfCpus": 4,
        "PhysicalCoresPerCpu": 2
    },
    "CpuLoadAverage": {
        "OneMinute": 2.24,
        "FiveMinutes": 0.97,
        "FifteenMinutes": 0.74
    },
    "Gpus": [
        {
            "Vendor": "NVIDIA Corporation",
            "ModelName": "GM204GL [Tesla M60]"
        }
    ],
    "LoggedInUsers": [
        {
            "Username" : "user45687"
        },
        {
            "Username" : "user789"
        }
    ]
},
"Tags": [
    {
        "Key": "size",
        "Value": "big"
    },
    {
        "Key": "dcv:os-family",
        "Value": "linux"
    },
    {
        "Key": "dcv:max-virtual-sessions",
        "Value": "10"
    },
    {
        "Key": "color",
        "Value": "blue"
    }
]
```

```
}  
  ]  
}
```

DescribeSessions

Describe una o varias sesiones de Amazon DCV.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

SessionIds

ID de la sesión que se va a describir.

Tipo: cadena

Requerido: no

NextToken

Token que se utilizará para recuperar la siguiente página de resultados.

Tipo: cadena

Requerido: no

Filters

Filtros adicionales para aplicar a la solicitud. Los filtros compatibles son:

- tag:key: etiquetas asignadas a la sesión.
- owner: propietario de la sesión.

Tipo: cadena

Requerido: no

Parámetros de respuesta

Id

ID exclusivo de la sesión.

Name

Nombre de la sesión.

Owner

Nombre del propietario de la sesión.

Server

Información sobre el servidor en el que se ejecuta la sesión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Ip

Dirección IP del host del servidor Amazon DCV.

Hostname

Nombre del host del servidor Amazon DCV.

Port

Puerto a través del cual el servidor Amazon DCV se comunica con clientes Amazon DCV.

Endpoints

Información sobre los puntos de conexión del servidor Amazon DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

Puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Los valores posibles son:

- HTTP: el punto de conexión utiliza el protocolo WebSocket (TCP).
- QUIC: el punto de conexión utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del punto de conexión del servidor. Disponible solo para el protocolo HTTP.

Tags

Etiquetas asignadas al servicio. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Type

Tipo de sesión.

State

Estado actual de la sesión. Los valores posibles son los siguientes:

- CREATING: el broker está creando la sesión.
- READY: la sesión está lista para aceptar conexiones de clientes.
- DELETING: se está eliminando la sesión.
- DELETED: se ha eliminado la sesión.
- UNKNOWN: no se puede determinar el estado de la sesión. Es posible que el agente y el broker no puedan comunicarse.

Substate

Subestado actual de la sesión. Los valores posibles son los siguientes:

- **SESSION_PLACING**: la sesión está esperando a ser colocada en un servidor DCV disponible.
- **PENDING_PREPARATION**: la sesión se ha creado pero no se puede utilizar; está vinculada a un servidor DCV.

CreationTime

Fecha y hora en que se creó la sesión.

LastDisconnectionTime

Fecha y hora de la última desconexión del cliente.

NumOfConnections

Número de conexiones de cliente activas.

StorageRoot

Especifica la ruta a la carpeta usada para el almacenamiento de sesión. Para obtener más información sobre el almacenamiento de sesiones de Amazon DCV, consulte [Habilitación de almacenamiento de sesión](#) en la Guía del administrador de Amazon DCV.

Tipo: cadena

Requerido: no

Ejemplo

Python

Solicitud

El siguiente ejemplo describe las sesiones que son propiedad de user1 y tienen una etiqueta de os=windows.

```
from swagger_client.models.describe_sessions_request_data import
    DescribeSessionsRequestData
from swagger_client.models.key_value_pair import KeyValuePair

def get_sessions_api():
```

```

    api_instance =
swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def describe_sessions(session_ids=None, next_token=None, tags=None, owner=None):
    filters = list()
    if tags:
        for tag in tags:
            filter_key_value_pair = KeyValuePair(key='tag:' + tag['Key'],
value=tag['Value'])
            filters.append(filter_key_value_pair)
    if owner:
        filter_key_value_pair = KeyValuePair(key='owner', value=owner)
        filters.append(filter_key_value_pair)

    request = DescribeSessionsRequestData(session_ids=session_ids, filters=filters,
next_token=next_token)
    print('Describe Sessions Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.describe_sessions(body=request)
    print('Describe Sessions Response', api_response)

def main():
    describe_sessions(
        owner='user1',
        tags=[{'Key': 'os', 'Value': 'windows'}])

```

Respuesta

A continuación se muestra la salida de ejemplo.

```

{
  "Sessions": [
    {
      "Id": "SessionId1897",
      "Name": "a session name",
      "Owner": "an owner 1890",
      "Server": {
        "Ip": "1.1.1.123",
        "Hostname": "server hostname",
        "Port": "1222",
        "Endpoints": [
          {

```

```
        "IpAddress": "x.x.x.x",
        "Port": 8443,
        "WebUrlPath": "/",
        "Protocol": "HTTP"
    },
    {
        "IpAddress": "x.x.x.x",
        "Port": 9443,
        "WebUrlPath": "/",
        "Protocol": "HTTP"
    },
    {
        "IpAddress": "x.x.x.x",
        "Port": 8443,
        "WebUrlPath": "",
        "Protocol": "QUIC"
    }
],
"Tags": [
    {
        "Key": "os",
        "Value": "windows"
    },
    {
        "Key": "ram",
        "Value": "4gb"
    }
]
},
"Type": "VIRTUAL",
"State": "READY",
"CreationTime": "2020-10-06T10:15:31.633Z",
"LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
"NumOfConnections": 2,
"StorageRoot" : "/storage/root"
},
{
    "Id": "SessionId1895",
    "Name": "a session name",
    "Owner": "an owner 1890",
    "Server": {
        "Ip": "1.1.1.123",
        "Hostname": "server hostname",
        "Port": "1222",
```

```
    "Endpoints": [
      {
        "IpAddress": "x.x.x.x",
        "Port": 8443,
        "WebUrlPath": "/",
        "Protocol": "HTTP"
      },
      {
        "IpAddress": "x.x.x.x",
        "Port": 9443,
        "WebUrlPath": "/",
        "Protocol": "HTTP"
      },
      {
        "IpAddress": "x.x.x.x",
        "Port": 8443,
        "WebUrlPath": "",
        "Protocol": "QUIC"
      }
    ],
    "Tags": [
      {
        "Key": "os",
        "Value": "windows"
      },
      {
        "Key": "ram",
        "Value": "4gb"
      }
    ]
  },
  "Type": "VIRTUAL",
  "State": "DELETING",
  "CreationTime": "2020-10-06T10:15:31.633Z",
  "LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
  "NumOfConnections": 2,
  "StorageRoot" : "/storage/root"
}
]
```

DeleteSessions

Elimina la sesión de Amazon DCV especificada y la elimina de la memoria caché del broker.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

SessionId

ID de la sesión que se va a eliminar.

Tipo: cadena

Obligatorio: sí

Owner

Propietario de la sesión que se va a eliminar.

Tipo: cadena

Obligatorio: sí

Force

Elimina una sesión de la memoria caché del broker al intentar eliminarla del servidor Amazon DCV. Resulta útil para eliminar las sesiones desactualizadas de la caché del broker. Por ejemplo, si se ha detenido un servidor Amazon DCV, pero las sesiones siguen registradas en el broker, utilice este indicador para purgar las sesiones de la caché del broker.

Tenga en cuenta que si la sesión sigue activa, el broker la volverá a almacenar en caché.

Valores válidos: `true` | `false`

Tipo: Booleano

Requerido: no

Parámetros de respuesta

SessionId

El ID de la sesión

State

Solo se devuelve si las sesiones se eliminaron correctamente. Indica el estado actual de la sesión. Si la solicitud se completa correctamente, la sesión pasa al estado DELETING. Puede que transcurran unos minutos hasta que se elimine la sesión. Cuando se elimina, el estado pasa de DELETING a DELETED.

FailureReason

Solo se devuelve si no se han podido eliminar algunas sesiones. Indica por qué no se pudo eliminar la sesión.

Ejemplo

Python

Solicitud

En el siguiente ejemplo, se eliminan dos sesiones: una sesión con un identificador de SessionId123 que es propiedad de user1 y una sesión con un identificador de SessionIdabc que es propiedad de user99.

```
from swagger_client.models.delete_session_request_data import
    DeleteSessionRequestData

def get_sessions_api():
    api_instance =
    swagger_client.SessionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def delete_sessions(sessions_to_delete, force=False):
    delete_sessions_request = list()
    for session_id, owner in sessions_to_delete:
        a_request = DeleteSessionRequestData(session_id=session_id, owner=owner,
        force=force)
```

```
delete_sessions_request.append(a_request)

print('Delete Sessions Request:', delete_sessions_request)
api_instance = get_sessions_api()
api_response = api_instance.delete_sessions(body=delete_sessions_request)
print('Delete Sessions Response', api_response)

def main():
    delete_sessions([('SessionId123', 'an owner user1'), ('SessionIdabc',
'user99')])
```

Respuesta

A continuación, se muestra el resultado de un ejemplo. `SessionId123` se eliminó correctamente, pero `SessionIdabc` no se ha podido eliminar.

```
{
  "RequestId": "10311636-df90-4cd1-bcf7-474e9675b7cd",
  "SuccessfulList": [
    {
      "SessionId": "SessionId123",
      "State": "DELETING"
    }
  ],
  "UnsuccessfulList": [
    {
      "SessionId": "SessionIdabc",
      "FailureReason": "The requested dcvSession does not exist"
    }
  ]
}
```

GetSessionConnectionData

Obtiene información de conexión para la conexión de un usuario específico a una sesión de Amazon DCV específica.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)

- [Información adicional](#)
- [Ejemplo](#)

Parámetros de solicitud

SessionId

ID de la sesión para la que se muestra información de conexión.

Tipo: cadena

Obligatorio: sí

User

Nombre del usuario para el que se muestra información de conexión.

Tipo: cadena

Obligatorio: sí

Parámetros de respuesta

Id

ID exclusivo de la sesión.

Name

Nombre de la sesión.

Owner

Nombre del propietario de la sesión.

Server

Información sobre el servidor en el que se ejecuta la sesión. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Ip

Dirección IP del host del servidor Amazon DCV.

Hostname

Nombre del host del servidor Amazon DCV.

Port

Puerto a través del cual el servidor Amazon DCV se comunica con clientes Amazon DCV.

Endpoints

Información sobre los puntos de conexión del servidor Amazon DCV. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

IpAddress

Dirección IP del punto de conexión del servidor.

Port

Puerto del punto de conexión del servidor.

Protocol

Protocolo utilizado por el punto de conexión del servidor. Los valores posibles son:

- HTTP: el punto de conexión utiliza el protocolo WebSocket (TCP).
- QUIC: el punto de conexión utiliza el protocolo QUIC (UDP).

WebUrlPath

La ruta URL web del punto de conexión del servidor. Disponible solo para el protocolo HTTP.

WebUrlPath

Ruta del archivo de configuración del servidor Amazon DCV.

Tags

Etiquetas asignadas al servicio. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Key

La clave de la etiqueta.

Value

El valor de la etiqueta.

Type

Tipo de sesión.

State

Estado actual de la sesión. Los valores posibles son los siguientes:

- **CREATING**: el broker está creando la sesión.
- **READY**: la sesión está lista para aceptar conexiones de clientes.
- **DELETING**: se está eliminando la sesión.
- **DELETED**: se ha eliminado la sesión.
- **UNKNOWN**: no se puede determinar el estado de la sesión. Es posible que el agente y el broker no puedan comunicarse.

CreationTime

Fecha y hora en que se creó la sesión.

LastDisconnectionTime

Fecha y hora de la última desconexión del cliente.

NumOfConnections

Número de conexiones simultáneas que el usuario tiene en la sesión.

ConnectionToken

Token de autenticación utilizado para conectarse a la sesión.

Información adicional

La información obtenida de esta API se puede pasar a un cliente Amazon DCV para conectarse a la sesión de Amazon DCV.

En el caso del cliente web Amazon DCV, puede crear una URL que se pueda abrir en el navegador. La URL tiene el siguiente formato:

```
https://{Ip}:{Port}{WebUrlPath}?authToken={ConnectionToken}#{SessionId}.
```

En el caso del cliente nativo Amazon DCV, puede crear una URL con el esquema `dcv://`. Cuando se instala el cliente nativo Amazon DCV, se registra en el sistema como controlador de las URL de `dcv://`. La URL tiene el siguiente formato:

```
dcv://{Ip}:{Port}{WebUrlPath}?authToken={ConnectionToken}#{SessionId}.
```

Note

Si utiliza Amazon EC2, la dirección IP debe ser la pública. Si su configuración tiene hosts Amazon DCV detrás de una puerta de enlace, especifique la dirección de la puerta de enlace en lugar de la que devuelve la API `SessionConnectionData`.

Ejemplo

Python

Solicitud

El siguiente ejemplo obtiene información de conexión de un usuario con un nombre de usuario de `user1` y una sesión con un ID de `sessionId12345`.

```
def get_session_connection_api():
    api_instance =
    swagger_client.GetSessionConnectionDataApi(swagger_client.ApiClient(get_client_configuratio
    set_request_headers(api_instance.api_client)
    return api_instance

def get_url_to_connect(api_response):
```

```
ip_address = api_response.session.server.ip
port = api_response.session.server.port
web_url_path = api_response.session.server.web_url_path
connection_token = api_response.connection_token
session_id = api_response.session.id
url = f'https://{ip_address}:{port}{web_url_path}?
authToken={connection_token}#{session_id}'
return url

def get_session_connection_data(session_id, user):
    api_response =
    get_session_connection_api().get_session_connection_data(session_id=session_id,
    user=user)
    url_to_connect = get_url_to_connect(api_response)
    print('Get Session Connection Data Response:', api_response)
    print('URL to connect: ', url_to_connect)

def main():
    get_session_connection_data('sessionId12345', 'user1')
```

Respuesta

A continuación se muestra la salida de ejemplo.

```
{
  "Session": {
    "Id": "sessionId12345",
    "Name": "a session name",
    "Owner": "an owner 1890",
    "Server": {
      "Ip": "1.1.1.123",
      "Hostname": "server hostname",
      "Port": "1222",
      "endpoints": [
        {
          "port": 8443,
          "web_url_path": "/",
          "protocol": "HTTP"
        },
        {
          "port": 9443,
          "web_url_path": "/",

```

```

        "protocol": "HTTP"
    },
    {
        "port": 8443,
        "web_url_path": "",
        "protocol": "QUIC"
    }
],
"WebUrlPath": "/path",
"Tags": [
    {
        "Key": "os",
        "Value": "windows"
    },
    {
        "Key": "ram",
        "Value": "4gb"
    }
]
},
"Type": "VIRTUAL",
"State": "UNKNOWN",
"CreationTime": "2020-10-06T10:15:31.633Z",
"LastDisconnectionTime": "2020-10-06T10:15:31.633Z",
"NumOfConnections": 2
},
"ConnectionToken":
"EXAMPLEi0iJm0WM1YTRhZi1jZmU0LTQ0ZjEtYjZlOC04ZjY0YjM4ZTE2ZDkiLCJ0eXAI0iJKV1QiLCJhbGciOiJSUz
tngiKXevUxhhJm3BPJYRs9NPE4GCJRTc13EXAMPLEIxNEPPh5IMcVmR0fU1WKPnry4ypPTp3rsZ7YWjCTSfs1GoN3R_
Kqtpd5GH0D-E8FwsedV-
Q2bRQ4y9y1q0MgFU4QjaSMypUuYR0YjkCaoainjmEZew4A33fG40wATrBvoivBiNwdNpytHX2CD0uk_k0k_DWeZjMvv9
h_GaMgHmltqBIA4jdPD7i0CmC2e7413KFy-
EQ4Ej1cM7RjLwhFuWpKWAVJxogJjYpfoKkaPo4KxvJjJIPYhksck1INQpe2W5rn1xCq7sC7ptcGw17DUobP7egRv9H37
hK1G4G8erHv19HlrTR9_c884fNrTCC8DvC062e4KYdLkAhhJmboN9CAGIGFyd2c1AY_CzzvDL0EXAMPLE"
}

```

GetSessionScreenshots

Obtiene capturas de pantalla de una o varias sesiones de Amazon DCV.

El tipo de archivo de imagen y la resolución de la captura de pantalla dependen de la configuración del broker de Session Manager. Para modificar el tipo de archivo de imagen, configure el parámetro

`session-screenshot-format`. Para modificar la resolución, configure los parámetros `session-screenshot-max-width` y `session-screenshot-max-height`. Para obtener más información, consulte [Archivo de configuración del broker](#) en la Guía del administrador de Amazon DCV Session Manager.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

SessionId

ID de la sesión de Amazon DCV de la que se obtiene la captura de pantalla.

Tipo: cadena

Obligatorio: sí

Parámetros de respuesta

RequestId

ID único de la solicitud.

SuccessfulList

Información sobre las capturas de pantalla correctas. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

SessionScreenshot

Información sobre las capturas de pantalla. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

SessionId

ID de la sesión de Amazon DCV desde la que se tomó la captura de pantalla.

Images

Información acerca de las imágenes. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

Format

Formato de los datos. Los valores posibles son: jpeg y png.

Data

La imagen de captura de pantalla está codificada en formato base64.

CreationTime

Fecha y hora en que se realizó la captura de pantalla.

Primary

Indica si la captura de pantalla es de la pantalla principal de la sesión de Amazon DCV.

UnsuccessfulList

Información sobre las capturas de pantalla incorrectas. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

GetSessionScreenshotRequestData

Solicitud original que falló.

SessionId

ID de la sesión de Amazon DCV desde la que se tomó la captura de pantalla.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

En el siguiente ejemplo, se obtienen capturas de pantalla de dos sesiones (`sessionId1` y `sessionId2`). El servidor `sessionId2` no existe y se produce un error.

```

from swagger_client.models.describe_servers_request_data import
    DescribeServersRequestData

def get_sessions_api():
    api_instance =
    swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def get_session_screenshots(session_ids):
    request = [GetSessionScreenshotRequestData(session_id=session_id) for session_id
in session_ids]
    print('Get Session Screenshots Request:', request)
    api_instance = get_sessions_api()
    api_response = api_instance.get_session_screenshots(body=request)
    print('Get Session Screenshots Response:', api_response)

def main():
    get_session_screenshots(["sessionId1", "sessionId2"])

```

Respuesta

A continuación se muestra la salida de ejemplo.

```

{
  "RequestId": "542735ef-f6ab-47d8-90e5-23df31d8d166",
  "SuccessfulList": [
    {
      "SessionScreenshot": {
        "SessionId": "sessionId1",
        "Images": [
          {
            "Format": "png",
            "Data": "iVBORw0KGgoAAAANSUgAAAEEXAMPLE",
            "CreationTime": "2021-03-30T15:47:06.822Z",
            "Primary": true
          }
        ]
      }
    }
  ],
  "UnsuccessfulList": [
    {

```

```
    "GetSessionScreenshotRequestData": {  
        "SessionId": "sessionId2"  
    },  
    "FailureReason": "Dcv session not found."  
  }  
]  
}
```

OpenServers

Abre uno o varios servidores Amazon DCV. Para crear sesiones de Amazon DCV en un servidor Amazon DCV debe cambiar el estado del servidor a abierto. Una vez abierto el servidor Amazon DCV, puede crear sesiones Amazon DCV en el servidor.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

ServerId

ID del servidor que se va a abrir.

Tipo: cadena

Obligatorio: sí

Parámetros de respuesta

RequestId

ID único de la solicitud.

SuccessfulList

Información sobre los servidores Amazon DCV que se abrieron correctamente. Esta estructura de datos incluye el siguiente parámetro de respuesta anidado:

ServerId

ID del servidor que se abrió correctamente.

UnsuccessfulList

Información sobre los servidores Amazon DCV que no se han podido abrir. Esta estructura de datos incluye los siguientes parámetros de respuesta anidados:

OpenServerRequestData

Información sobre la solicitud original que falló. Esta estructura de datos incluye el siguiente parámetro de respuesta anidado:

ServerId

ID del servidor Amazon DCV que no se ha podido abrir.

FailureCode

Código del error.

FailureReason

El motivo del error.

Ejemplo

Python

Solicitud

El siguiente ejemplo abre dos servidores Amazon DCV (`serverId1` y `serverId2`).

```
from swagger_client.models import OpenServerRequestData

def get_servers_api():
    api_instance =
    swagger_client.ServersApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def open_servers(server_ids):
```

```
request = [OpenServerRequestData(server_id=server_id) for server_id in
server_ids]
print('Open Servers Request:', request)
api_instance = get_servers_api()
api_response = api_instance.open_servers(body=request)
print('Open Servers Response:', api_response)

def main():
    open_servers(["serverId1", "serverId2"])
```

Respuesta

A continuación se muestra la salida de ejemplo.

```
{
  "RequestId": "1e64830f-0a27-41bf-8147-0f3411791b64",
  "SuccessfulList": [
    {
      "ServerId": "serverId1"
    }
  ],
  "UnsuccessfulList": [
    {
      "OpenServerRequestData": {
        "ServerId": "serverId2"
      },
      "FailureCode": "DCV_SERVER_NOT_FOUND",
      "FailureReason": "Dcv server not found."
    }
  ]
}
```

UpdateSessionPermissions

Actualiza los permisos de usuario para una sesión de Amazon DCV específica.

Temas

- [Parámetros de solicitud](#)
- [Parámetros de respuesta](#)
- [Ejemplo](#)

Parámetros de solicitud

SessionId

ID de la sesión cuyos permisos se van a cambiar.

Tipo: cadena

Obligatorio: sí

Owner

Propietario de la sesión cuyos permisos se van a cambiar.

Tipo: cadena

Obligatorio: sí

PermissionFile

Contenido codificado en Base64 del archivo de permisos. Para obtener más información, consulte [Configuración de la autorización de Amazon DCV](#) en la Guía del administrador de Amazon DCV.

Tipo: cadena

Obligatorio: sí

Parámetros de respuesta

SessionId

El ID de la sesión.

Ejemplo

Python

Solicitud

En el ejemplo siguiente se establecen nuevos permisos para una sesión con un identificador de sesión de `SessionId1897`.

```

from swagger_client.models.update_session_permissions_request_data import
    UpdateSessionPermissionsRequestData

def get_session_permissions_api():
    api_instance =
    swagger_client.SessionPermissionsApi(swagger_client.ApiClient(get_client_configuration()))
    set_request_headers(api_instance.api_client)
    return api_instance

def update_session_permissions(session_permissions_to_update):
    update_session_permissions_request = list()
    for session_id, owner, permissions_base64_encoded in
    session_permissions_to_update:
        a_request = UpdateSessionPermissionsRequestData(
            session_id=session_id, owner=owner,
            permissions_file=permissions_base64_encoded)
        update_session_permissions_request.append(a_request)
    print('Update Session Permissions Request:', update_session_permissions_request)
    api_instance = get_session_permissions_api()
    api_response =
    api_instance.update_session_permissions(body=update_session_permissions_request)
    print('Update Session Permissions Response:', api_response)

def main():
    update_session_permissions([('SessionId1897', 'an owner 1890',
    'file_base64_encoded')])

```

Respuesta

A continuación se muestra la salida de ejemplo.

```

{
  'request_id': 'd68ebf66-4022-42b5-ba65-99f89b18c341',
  'successful_list': [
    {
      session_id: 'SessionId1897'
    }
  ],
  'unsuccessful_list': []
}

```

Notas de la versión e historial de revisión de Amazon DCV Session Manager

Esta página recoge las notas de la versión y el historial de revisión de Amazon DCV Session Manager.

Temas

- [Notas de la versión de Amazon DCV Session Manager](#)
- [Historial de documentos](#)

Notas de la versión de Amazon DCV Session Manager

Esta sección proporciona una descripción general de las principales actualizaciones, lanzamientos de características y correcciones de errores de Amazon DCV Session Manager. Todas las actualizaciones están organizadas por fecha de lanzamiento. Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envían.

Temas

- [2024.0-457: 1 de octubre de 2024](#)
- [2023.1-17652: 1 de agosto de 2024](#)
- [2023.1-16388: 26 de junio de 2024](#)
- [2023.1— 9 de noviembre de 2023](#)
- [2023.0-15065: 4 de mayo de 2023](#)
- [2023.0-14852— 28 de marzo de 2023](#)
- [2022.2-13907— 11 de noviembre de 2022](#)
- [2022.1-13067— 29 de junio de 2022](#)
- [2022.0-11952— 23 de febrero de 2022](#)
- [2021.3-11591— 20 de diciembre de 2021](#)
- [2021.2-11445— 18 de noviembre de 2021](#)
- [2021.2-11190— 11 de octubre de 2021](#)
- [2021.2-11042: 1 de septiembre de 2021](#)

- [2021.1-10557— 31 de mayo de 2021](#)
- [2021.0-10242— 12 de abril de 2021](#)
- [2020.2-9662— 04 de diciembre de 2020](#)
- [2020.2-9508— 11 de noviembre de 2020](#)

2024.0-457: 1 de octubre de 2024

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 457• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Cambió el nombre de NICE DCV a Amazon DCV.• Se ha agregado compatibilidad con Ubuntu 24.04.

2023.1-17652: 1 de agosto de 2024

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 426• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Correcciones de errores y mejoras de rendimiento.

2023.1-16388: 26 de junio de 2024

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 417• Agente: 748• CLI: 140	<ul style="list-style-type: none">• Se ha corregido un error que mostraba incorrectamente la memoria en TB, no en GB.• Correcciones de errores y mejoras de rendimiento.

2023.1— 9 de noviembre de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 410• Agente: 732• CLI: 140	<ul style="list-style-type: none">• Correcciones de errores y mejoras de rendimiento.

2023.0-15065: 4 de mayo de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 392• Agente: 675• CLI: 132	<ul style="list-style-type: none">• Se ha agregado compatibilidad con Red Hat Enterprise Linux 9, Rocky Linux 9 y CentOS Stream 9 en plataformas ARM.

2023.0-14852— 28 de marzo de 2023

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 392• Agente: 642• CLI: 132	<ul style="list-style-type: none">• Se ha agregado compatibilidad con Red Hat Enterprise Linux 9, Rocky Linux 9 y CentOS Stream 9.

2022.2-13907— 11 de noviembre de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 382• Agente: 612• CLI: 123	<ul style="list-style-type: none">• Se ha agregado un campo Substate en respuesta a DescribeSessions .• Se ha corregido un problema que podía provocar que la CLI no pudiera conectarse al broker en función de la URL utilizada.

2022.1-13067— 29 de junio de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 355• Agente: 592• CLI: 114	<ul style="list-style-type: none">• Se ha agregado compatibilidad para ejecutar el broker en instancias de AWS Graviton.• Se ha agregado compatibilidad con agentes y brokers para Ubuntu 22.04.

2022.0-11952— 23 de febrero de 2022

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 341• Agente: 520• CLI: 112	<ul style="list-style-type: none">• Se ha agregado la capacidad de rotación de registros al agente.• Se ha agregado un parámetro de configuración para establecer Java Home en el broker.• Se ha mejorado el vaciado de datos de la memoria caché al disco en el broker.• Validación de URL fija en la CLI.

2021.3-11591— 20 de diciembre de 2021

Números de compilación	Nuevas características
<ul style="list-style-type: none">• Broker: 307• Agente: 453• CLI: 92	<ul style="list-style-type: none">• Se ha agregado compatibilidad para la integración con la puerta de enlace de conexión Amazon DCV.• Se ha agregado compatibilidad con Ubuntu 18.04 y Ubuntu 20.04.

2021.2-11445— 18 de noviembre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 288• Agente: 413• CLI: 54	<ul style="list-style-type: none">• Se ha corregido un problema con la validación de los nombres de inicio de sesión que incluyen un dominio de Windows.

2021.2-11190— 11 de octubre de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none">• Broker: 254• Agente: 413• CLI: 54	<ul style="list-style-type: none">• Se ha corregido un problema en la interfaz de la línea de comandos que impedía iniciar sesiones de Windows.

2021.2-11042: 1 de septiembre de 2021

Números de compilación	Nuevas características	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 254 • Agente: 413 • CLI: 37 	<ul style="list-style-type: none"> • Amazon DCV Session Manager ofrece ahora compatibilidad con la interfaz de la línea de comandos (CLI). Puede crear y administrar sesiones de Amazon DCV en la CLI, en lugar de llamar a las API. • Amazon DCV Session Manager ha introducido persistencia de datos del broker. Para mayor disponibilidad, los agentes pueden conservar la información de estado del servidor en un almacén de datos externo y restaurarlos al iniciar. 	<ul style="list-style-type: none"> • Al registrar un servidor de autorización externo, ahora puede especificar el algoritmo que utiliza el servidor de autorización para firmar los Web Tokens con formato JSON. Con este cambio, puede utilizar Azure AD como servidor de autorización externo.

2021.1-10557— 31 de mayo de 2021

Números de compilación	Nuevas características	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 214 • Agente: 365 	<ul style="list-style-type: none"> • Amazon DCV Session Manager ha agregado compatibilidad con los parámetros de entrada pasados al archivo de ejecución automática en Linux. 	<ul style="list-style-type: none"> • Hemos solucionado un problema con el archivo de ejecución automática en Windows.

Números de compilación	Nuevas características	Cambios y correcciones de errores
	<ul style="list-style-type: none"> Las propiedades del servidor ahora se pueden pasar como requisitos a la API CreateSessions. 	

2021.0-10242— 12 de abril de 2021

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> Broker: 183 Agente: 318 	<ul style="list-style-type: none"> Amazon DCV Session Manager ha introducido las siguientes API nuevas: <ul style="list-style-type: none"> OpenServers CloseServers DescribeServers GetSessionScreenshots También ha introducido los siguientes parámetros de configuración nuevos: <ul style="list-style-type: none"> Parámetros del broker: <code>session-screenshot-max-width</code>, <code>session-screenshot-max-height</code>, <code>session-screenshot-format</code>, <code>create-sessions-queue-max-size</code> y <code>create-sessions-queue-max-time-seconds</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>. Parámetros del agente: <code>agent.autorun_folder</code>, <code>max_virtual_sessions</code> y <code>max_concurrent_sessions_per_user</code>.

2020.2-9662— 04 de diciembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 114 • Agente: 211 	<ul style="list-style-type: none"> • Hemos solucionado un problema con los certificados TLS generados automáticamente que impedían que el broker se iniciara.

2020.2-9508— 11 de noviembre de 2020

Números de compilación	Cambios y correcciones de errores
<ul style="list-style-type: none"> • Broker: 78 • Agente: 183 	<ul style="list-style-type: none"> • Versión inicial de Amazon DCV Session Manager.

Historial de documentos

En la siguiente tabla se describe la documentación de esta versión de Amazon DCV Session Manager.

Cambio	Descripción	Fecha
Amazon DCV versión 2024.0-457	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2024.0-457. Para obtener más información, consulte 2024.0-457: 1 de octubre de 2024 .	30 de septiembre de 2024
Amazon DCV versión 2023.1-17652	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2023.1-17652. Para obtener más información, consulte 2023.1-17652: 1 de agosto de 2024 .	1 de agosto de 2024

Cambio	Descripción	Fecha
Amazon DCV versión 2023.1-16388	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2023.1-16388. Para obtener más información, consulte 2023.1-16388: 26 de junio de 2024 .	26 de junio de 2024
Amazon DCV versión 2023.1	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2023.1. Para obtener más información, consulte 2023.1— 9 de noviembre de 2023 .	9 de noviembre de 2023
Amazon DCV versión 2023.0	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2023.0. Para obtener más información, consulte 2023.0-14852— 28 de marzo de 2023 .	28 de marzo de 2023
Amazon DCV versión 2022.2	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2022.2. Para obtener más información, consulte 2022.2-13907— 11 de noviembre de 2022 .	11 de noviembre de 2022
Amazon DCV versión 2022.1	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2022.1. Para obtener más información, consulte 2022.1-13067— 29 de junio de 2022 .	29 de junio de 2022
Amazon DCV versión 2022.0	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2022.0. Para obtener más información, consulte 2022.0-11952— 23 de febrero de 2022 .	23 de febrero de 2022

Cambio	Descripción	Fecha
Amazon DCV versión 2021.3	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2021.3. Para obtener más información, consulte 2021.3-11591— 20 de diciembre de 2021 .	20 de diciembre de 2021
Amazon DCV versión 2021.2	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2021.2. Para obtener más información, consulte 2021.2-11042: 1 de septiembre de 2021 .	01 de septiembre de 2021
Amazon DCV versión 2021.1	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2021.1. Para obtener más información, consulte 2021.1-10557— 31 de mayo de 2021 .	31 de mayo de 2021
Amazon DCV versión 2021.0	Amazon DCV Session Manager se ha actualizado para Amazon DCV 2021.0. Para obtener más información, consulte 2021.0-10242— 12 de abril de 2021 .	12 de abril de 2021
Versión inicial de Amazon DCV Session Manager	Primera publicación de este contenido.	11 de noviembre de 2020